

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

# **ОСНОВИ ОБЧИСЛЮВАЛЬНОЇ ГЕОМЕТРІЇ**

**Конспект лекцій дисципліни**

## **03.14 ОБЧИСЛЮВАЛЬНА ГЕОМЕТРІЯ ТА КОМП'ЮТЕРНА ГРАФІКА**

**для напряму підготовки**

**6.040301 – «Прикладна математика»**

*Рекомендовано  
вченою радою  
ФПМ НТУУ «КПІ»  
Протокол №3  
від "24" жовтня 2015 р.*

Київ  
«Просвіта»  
2015

**УДК 004.922 (075.8)**

**ББК 22.143я73**

**С40**

*Рекомендовано вченою радою ФПМ НТУУ «КПІ»  
Протокол №3 від "24" жовтня 2015 р.*

Укладачі

*Сирота С. В., к. т. н., доцент кафедри ПМА*  
*Ліскін В. О. асистент кафедри ПМА*

Відповідальний за випуск *Сирота С.В.,*

Рецензент д. т. н., проф. \_

**Сирота С. В.**

**С40** Основи обчислювальної геометрії : Конспект лекцій : дисципліни 03.14 «Обчислювальна геометрія та комп'ютерна графіка» для напряму підгот. 6.040301 "Приклад. математика" / Сирота С. В., Ліскін В. О. ; Нац. техн. ун-т України "Київ. політехн. ін-т". — Київ : Просвіта, 2015. — 32 с.  
ISBN 978-617-7010-09-7

Навчальне видання призначене для студентів спеціальності 6.040301 – «Прикладна математика» містить лекції, дають уявлення основні математичні та алгоритмічні прийоми моделювання сцен та відтворення їх зображень на моніторах.

**УДК 512.64(075.8)**  
**ББК 22.143я73**

ISBN 978-617-7010-09-7

(С) НТУУ «КПІ» 2015

## ВСТУП

Предметом навчальної дисципліни «Обчислювальна геометрія та комп'ютерна графіка» (ОГКГ) є програмне забезпечення, математичні основи та базові алгоритми комп'ютерної графіки. Без комп'ютерної графіки не обходиться жодна сучасна мультимедійна програма. Робота над графікою займає до 90% робочого часу програмістських колективів, які випускають програми масового використання. Сфера застосування комп'ютерної графіки дуже різноманітна.

Наведені лекції містять математичні основи розділів обчислювальної геометрії, які використовуються в комп'ютерній графіці і стосуються моделювання сцени: поняття про однорідні координати, афінні перетворення на площині і в просторі, види проекцій та проектування, Також викладені математичні основи рендерингу і фільтрації.

Посібник складено відповідно до програми навчальної дисципліни «Обчислювальна геометрія та комп'ютерна графіка» напряму підготовки 6.040301 – «Прикладна математика». Видання призначене для самостійної роботи над лекційним матеріалом дисципліни.

# 1. АФІННІ ГЕОМЕТРИЧНІ ПЕРЕТВОРЕННЯ

## 1.1 Двовимірні афінні перетворення

Частковий випадок афінних перетворень, які найчастіше використовуються в комп'ютерній графіці може бути представлено комбінацією зсуву повороту і масштабування.

Точку з площини  $XY$  можна перенести у нове положення шляхом додавання до координат цієї точки зміщення.

Для кожної точки  $P(x, y)$ , яка переміщується у нову точку  $P'(x', y')$ , зсуючись на  $D_x$  одиниць паралельно вісі  $X$  та на  $D_y$  одиниць паралельно вісі  $Y$ , (рис.1.1-а) можна записати рівняння

$$x' = x + D_x, \quad y' = y + D_y. \quad (1)$$

Визначаючи вектори-рядки

$$P = (x \quad y), \quad P' = (x' \quad y'), \quad T = (D_x \quad D_y)$$

можна переписати рівняння (1) у векторній формі

$$(x' \quad y') = (x \quad y) + (D_x \quad D_y), \quad (2)$$

$$\text{або } P' = P + T. \quad (3)$$

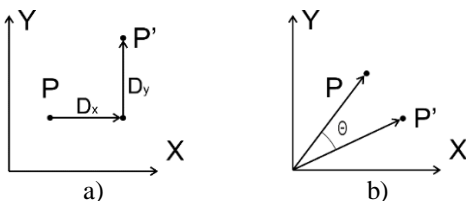


Рис 1.1 – елементарні перетворення. а-перенос, б- поворот

Об'єкт можна перенести, застосовуючи рівняння (1) до кожної його точки. Якщо об'єкт складається з відрізків прямих, можна перенести крайні точки відрізків і накреслити новий відрізок. Це твердження справедливе також для масштабування та обертання.

Об'єкт можна масштабувати (розтягнути) у  $S_x$  разів вздовж вісі  $X$  та у  $S_y$  разів вздовж вісі  $Y$ , отримавши у результаті нову точку шляхом множення

$$x' = x \cdot S_x, \quad y' = y \cdot S_y. \quad (4)$$

Визначаючи  $S$  як  $\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$ , можна записати у матричній формі

$$(x' \quad y') = (x \quad y) \cdot \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}, \quad (5)$$

$$P' = P \cdot S. \quad (6)$$

Описане масштабування виконується відносно початку координат. Якщо  $S_x \neq S_y$ , то це неоднорідне масштабування. Однорідне масштабування, для якого  $S_x = S_y$ , на відміну від неоднорідного, не змінює пропорції.

Точки можна обернути на кут  $\theta$  відносно початку координат (рис. 1.1-b)

$$\begin{aligned}x' &= x \cdot \cos \theta - y \cdot \sin \theta, \\y' &= x \cdot \sin \theta + y \cdot \cos \theta.\end{aligned}\tag{7}$$

В матричній формі ми маємо

$$\begin{pmatrix} x' & y' \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix},\tag{8}$$

$$P' = P \cdot R,\tag{9}$$

де  $R$  - матриця обертання з виразу (8).

Як і у випадку масштабування, обертання виконується відносно початку координат. Додатними вважаються кути, що вимірюються проти руху годинної стрілки від  $X$  до  $Y$ .

## 1.2 Однорідні координати та

### матричне представлення двовимірних перетворень

Перетворення перенесення, масштабування та обертання у матричній формі мають вигляд:

$$P' = P + T,\tag{1}$$

$$P' = P \cdot S,\tag{2}$$

$$P' = P \cdot R.\tag{3}$$

Якщо ми визначимо точки у однорідних координатах, то усі три види перетворень можна виконати за допомогою множення.

У однорідних координатах точка  $P(x, y)$  записується як  $P(W \cdot x, W \cdot y, W)$  для будь-якого масштабного множника  $W \neq 0$ . Якщо для точки задане її представлення у однорідних координатах  $P(X, Y, W)$ , то можна знайти її двовимірні декартові координати як  $x = \frac{X}{W}$ ,  $y = \frac{Y}{W}$ . Ми будемо приймати  $W = 1$ , тому ділення не буде потрібне.

Точки тепер описуються тривимірними векторами-рядками, тому матриці перетворення, на які множиться вектор координат точки, щоб отримати вектор координат іншої точки, повинні мати розмір  $3 \times 3$ .

Рівняння перенесення (3) записується у вигляді матриці перетворення однорідних координат наступним чином:

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix}\tag{4}$$

або

$$P' = P \cdot T(D_x, D_y), \quad (5)$$

де

$$T(D_x, D_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_x & D_y & 1 \end{bmatrix} \quad (6)$$

Що буде, якщо точку  $P$  перенести у точку  $P'$  на відстань  $(D_{x1}, D_{y1})$ , а потім у  $P''$  на відстань  $(D_{x2}, D_{y2})$ ? Інтуїтивно очікуваний результат у цьому випадку є сумарним перенесенням на відстань  $(D_{x1} + D_{x2}, D_{y1} + D_{y2})$ . Щоб довести це, запишемо дані у вигляді

$$P' = P \cdot T(D_{x1}, D_{y1}), \quad (7)$$

$$P'' = P' \cdot T(D_{x2}, D_{y2}). \quad (8)$$

Підставляючи (15) у (16), отримаємо

$$P'' = (P \cdot T(D_{x1}, D_{y1})) \cdot T(D_{x2}, D_{y2}) = P \cdot (T(D_{x1}, D_{y1}) \cdot T(D_{x2}, D_{y2})). \quad (9)$$

Матричний добуток  $T(D_{x1}, D_{y1}) \cdot T(D_{x2}, D_{y2})$  є

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_{x1} & D_{y1} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_{x2} & D_{y2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ D_{x1} + D_{x2} & D_{y1} + D_{y2} & 1 \end{bmatrix} \quad (10)$$

Дійсно, результуюче перенесення є  $(D_{x1} + D_{x2}, D_{y1} + D_{y2})$ .

Матричний добуток ми будемо називати КОМПОЗИЦІЄЮ матриць.

Рівняння масштабування (6) у матричному вигляді записується у вигляді:

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Визначаючи

$$S(S_x, S_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

маємо

$$P' = P \cdot S(S_x, S_y). \quad (13)$$

Так само як послідовні перенесення адитивні, можна очікувати, що послідовні масштабування будуть мультиплікативні. Тоді, якщо задані наступні перетворення

$$P' = P \cdot S(S_{x1}, S_{y1}), \quad (14)$$

$$P'' = P' \cdot S(S_{x2}, S_{y2}). \quad (15)$$

то, підставляючи (22) у (23) отримаємо

$$P'' = (P \cdot S(S_{x1}, S_{y1})) \cdot S(S_{x2}, S_{y2}) = P \cdot [S(S_{x1}, S_{y1}) \cdot S(S_{x2}, S_{y2})] \quad (16)$$

Матричний добуток  $S(S_{x1}, S_{y1}) \cdot S(S_{x2}, S_{y2})$  можна записати як

$$\begin{bmatrix} S_{x1} & 0 & 0 \\ 0 & S_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_{x2} & 0 & 0 \\ 0 & S_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} S_{x1} \cdot S_{x2} & 0 & 0 \\ 0 & S_{y1} \cdot S_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Таким чином масштабування дійсно мультиплікативні.

Рівняння обертання (9) можна представити у вигляді:

$$\begin{pmatrix} x' & y' & 1 \end{pmatrix} = \begin{pmatrix} x & y & 1 \end{pmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (18)$$

Вважаючи

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

маємо

$$P' = P \cdot R(\theta). \quad (20)$$

Аналогічно масштабуванню можна показати, що два послідовних обертання мультиплікативні.

### 1.3. Композиція двовимірних перетворень

Покажемо, як можна використовувати композицію перетворень для об'єднання фундаментальних матриць перетворень  $R$ ,  $S$ ,  $T$  з метою отримання бажаних загальних результатів. Основна перевага об'єднаних перетворень полягає у тому, що до точки більш ефективно застосувати одне результуюче перетворення, ніж послідовність перетворень.

Розглянемо, наприклад, обертання об'єкта навколо довільної точки  $P_1$ . Розіб'ємо це перетворення на три елементарних:

1. Перенесення, під час якого точка  $P_1$  переміщується у початок координат.
2. Обертання відносно початку координат.
3. Перенесення, під час якого точка з початку координат повертається у попереднє положення  $P_1$ .

Перше перенесення виконується на  $(-x_1, -y_1)$ , в той час як друге – на  $(x_2, y_2)$ . Результуюче перетворення має вигляд

$$\begin{aligned}
& \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_1 & -y_1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_1 & y_1 & 1 \end{bmatrix} = \\
& = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ x_1 \cdot (1 - \cos \theta) + y_1 \cdot \sin \theta & y_1 \cdot (1 - \cos \theta) - x_1 \cdot \sin \theta & 1 \end{bmatrix}
\end{aligned} \tag{29}$$

Ця композиція перетворень, отримана шляхом множення матриць, є прикладом того, як застосування однорідних координат спрощує задачу.

Нехай потрібно масштабувати, обернути і розташувати у потрібному місці об'єкт з відрізків прямих (наприклад, будиночок), де центром обертання та масштабування є точка  $P_1$ . Послідовність перетворень полягає у перенесенні точки  $P_1$  у початок координат, проведення масштабування та обертання, потім перенесенні з початку координат у точку  $P_2$ , в який будиночок має опинитися. Може бути записана матриця результуючого перетворення:

$$T(-x_1, -y_1) \cdot S(S_x, S_y) \cdot R(\theta) \cdot T(x_2, y_2) \tag{30}$$

Якщо відомо, що  $M_1$  та  $M_2$  – елементарні перетворення, то в яких випадках вони комутативні? Відомо, що в загальному випадку множення матриць не комутативне. Але неважно показати, що у наступних випадках комутативність має місце.

Перенесення – Перенесення

Масштабування – Масштабування

Обертання – Обертання

Однорідне масштабування – Обертання

У цих випадках можна не турбуватися про послідовність множення матриць.

#### 1.4. Підвищення ефективності обчислень

Композиція найбільш загального вигляду, що складається з операцій  $R$ ,  $S$ ,  $T$  має матрицю вигляду

$$M = \begin{vmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ t_x & t_y & 1 \end{vmatrix} \tag{1}$$

Структура останнього стовпчика матриці дозволяє спростити перетворення.

$$x' = x \cdot r_{11} + y \cdot r_{21} + t_x,$$

$$y' = x \cdot r_{12} + y \cdot r_{22} + t_y. \tag{2}$$



Це суттєво прискорює процес.

Таким чином, незважаючи на те, що матриці  $3 \times 3$  зручні і корисні для суміщення перетворень, у програмі доцільно використовувати результуючу матрицю з урахуванням її структури.

Іншою областю застосування спрощеного перетворення є генерація послідовних кадрів-зображень об'єкта, коли кожний наступний відрізняється від попереднього обертанням на декілька градусів. Спростити рівняння (7) можна, скориставшись тим, що кут  $\theta$  малий і  $\cos\theta$  приблизно дорівнює 1. Рівняння (7) тоді приймає вигляд

$$\begin{aligned}x' &= x - y \cdot \sin \theta, \\y' &= x \cdot \sin \theta + y.\end{aligned}\tag{3}$$

Можна отримати ще кращу апроксимацію, якщо у другому рівнянні використати  $x'$  замість  $x$ :

$$\begin{aligned}x' &= x - y \cdot \sin \theta, \\y' &= x' \cdot \sin \theta + y.\end{aligned}\tag{4}$$

Визначник відповідної матриці розміром  $2 \times 2$  тепер дорівнює 1, як і визначник усіх матриць обертання.

Треба лише врахувати, що у разі багаторазового використання формул (3) або (4) похибка буде накопичуватись. Ці труднощі можна подолати, якщо зберігати початковий список координат  $(x, y)$ . Після кожного обертання на  $360^\circ$  отримані у результаті дані відкидаються, а для наступного обертання беруться початкові дані.

### 1.5. Тривимірні афінні геометричні перетворення

Аналогічно тому, як двовимірні перетворення описуються матрицею  $3 \times 3$ , трьохвимірні перетворення можуть бути представлені у вигляді матриць розміром  $4 \times 4$ . Тоді трьохвимірна точка  $(x, y, z)$  записується у однорідних координатах як  $(W \cdot x, W \cdot y, W \cdot z, W)$ , де  $W \neq 0$ .

Для геометричних перетворень ми будемо розглядати *правосторонню систему координат*. Додатними будемо вважати ті обертання, при яких, якщо дивитися з кінця додатної півосі у початок координат, обертання на  $90^\circ$  проти годинникової стрілки буде переводити одну додатну піввісь у другу.

*Тривимірне перенесення* є простим розширенням двовимірного:

$$T(D_x, D_y, D_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ D_x & D_y & D_z & 1 \end{bmatrix}\tag{1}$$

$$\text{тобто } \begin{pmatrix} x & y & z & 1 \end{pmatrix} \cdot T(D_x, D_y, D_z) = \begin{pmatrix} x + D_x & y + D_y & z + D_z & 1 \end{pmatrix}\tag{2}$$

Масштабування розширюється аналогічним чином:

$$S(S_x, S_y, S_z) = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (37)$$

$$\text{Дійсно, } (x \ y \ z \ 1) \cdot S(S_x, S_y, S_z) = (S_x \cdot x \ S_y \cdot y \ S_z \cdot z \ 1). \quad (38)$$

Двовимірне обертання, що описане у двовимірних геометричних перетвореннях є одночасно тривимірним обертанням навколо осі Z. У трьохвимірному просторі обертання навколо осі Z описується виразом:

$$R_Z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (39)$$

Це легко перевірити: у результаті обертання на  $90^\circ$  вектора  $(1 \ 0 \ 0 \ 1)$ , що є одиничним вектором осі X, повинен створитися одиничний вектор  $(0 \ 1 \ 0 \ 1)$  осі Y.

$$\text{Обчислюючи добуток } (1 \ 0 \ 0 \ 1) \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (40)$$

отримаємо передбачений результат  $(0 \ 1 \ 0 \ 1)$ .

Матриця обертання навколо осі X має вигляд

$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (41)$$

Матриця обертання навколо осі Y записується у вигляді

$$R_Y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (42)$$

Можна перемножити довільну кількість матриць обертання масштабування та переносу і результат завжди матиме вигляд:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} \quad (43)$$

Верхня ліва підматриця  $R \ 3 \times 3$  буде описувати сумарні обертання та масштабування, у той час, як підматриця  $T$  буде задавати наступне сукупне перенесення. З точки зору швидкодії проведення обчислень корисніше виконувати перетворення явно.

$$(x' \ y' \ z') = (x \ y \ z) \cdot R + T. \quad (44)$$

Композиція тривимірних перетворень виконується аналогічно композиції двовимірних.

## 2. ПРЕДСТАВЛЕННЯ ПРОСТОРОВИХ ФОРМ

### 2.1. Способи представлення

У багатьох областях машинної графіки виникає потреба в представленні тривимірних форм: при проектуванні, при відновленні тривимірних тіл за зображеннями їх поперечних перерізів, побудованих за допомогою машинної графіки і багатьох інших. Необхідність описувати просторові об'єкти виникає в двох ситуаціях:

1) Коли ми хочемо описати вже існуючий об'єкт. У загальному випадку реально існуючий об'єкт не може повністю відповідати своєму представленню. Для цього у гіршому випадку знадобиться нескінченна кількість трійок  $(x, y, z)$  – по одній для кожної точки поверхні об'єкту. У кращому випадку об'єкт вдається представити комбінацією математичних поверхонь (площини, сфери та ін.). Як правило, ми вибираємо деяку множину точок на об'єкті і вимагаємо, щоб представлення об'єкту із заданою точністю відповідало самому об'єкту тільки в цих вибраних точках. Представлення існуючих поверхонь, таким чином, повинно задовольняти цим обмеженням.

2) Автоматизоване проектування. В цьому випадку немає заздалегідь існуючої моделі об'єкту що проектується: конструктор, керуючись, можливо, лише приблизним ескізом, створює просторову форму в інтерактивному режимі. Використовуване представлення має бути зручне для роботи в режимі діалогу, щоб поверхню можна було легко привести до бажаної форми.

Найбільш поширеними способами представлення тривимірних поверхонь в просторі є: полігональні сітки і параметричні бікубічні шматки.

Параметричні бікубічні шматки описують координати точок на викривленій поверхні за допомогою трьох рівнянь (по одному для  $x$ ,  $y$ ,  $z$ ). Кожне рівняння

має дві змінні (параметри), при чому показники степені при них не вищі за третій (звідси назва бікубічний). Межами шматків є параметричні бікубічні криві. Для представлення поверхонь із заданою точністю потрібно значно менше число бікубічних шматків, чим при апроксимації полігональною сіткою. Проте алгоритми для роботи з бікубічними об'єктами істотно складніші за алгоритми, що мають справу з багатокутниками.

При використанні обох методів тривимірне тіло представляється у вигляді замкнутої поверхні.

У відмінності від цього при моделюванні суцільних тіл мають справу з твердими об'єктами. Як правило в багатьох 3d редакторах і бібліотеках використовують примітиви такі як, як куб, конус, сфера, циліндри, тор, труба, бокс, площина і чомусь чайник, які можна об'єднувати і віднімати, утворюючи різні форми.

Для опису поверхонь спочатку розглянемо представлення просторових кривих, а потім зробимо математичне узагальнення від просторових кривих до тривимірних криволінійних поверхонь.

## 2.2.Параметричні бікубічні криві

Основна мета – описати криву за допомогою послідовності сегментів кривої. Як правило всі векторні редактори мають схожий інтерфейс редагування кривих. Щоб змінити форму кривої користувач за допомогою спеціального інструменту може змінювати розташування вузла і кінці дотичних до нього векторів. Щоб крива не мала зламів у точках з'єднання сегментів кривої їх дотичні вектори мають бути колінеарні. (Рис 2.1).



Рис. 2.1 – редагування кривих в CorelDraw

Нижче для опису ми користуватимемося загальним позначенням  $C^{(i)}$  -неперервності. Говорять, що криві  $C^{(0)}$  - неперервні, якщо вони не мають розривів, і криві  $C^{(1)}$  - неперервні, якщо крім того, неперервні їх дотичні (похідні).

Існують два основні способи представлення кривих: за допомогою функцій змінних  $x, y, z$  і функцій деякого параметра  $t$ . У першому випадку, щоб визначити точки  $(x, y, z)$  на кривій, доводиться мати справу з функціями виду

$$x = x, y = f(x), z = g(x). \quad (1)$$

Це представлення спричиняє багато ускладнень: в деякій точці кривий тангенс кута нахилу може виявитися рівним нескінченності. Непростою також є перевірка, чи належить точка деякому сегменту кривої, заданому початковою і кінцевою точками.

У разі параметричного представлення тривимірних кривих ці проблеми істотно спрощуються, оскільки при такому представленні легко описувати замкнуті багатозначні функції і, крім того, замість тангенсів кутів нахилу використовуються дотичні вектори, які ніколи не бувають нескінченними.

Параметричною кубічною кривою є крива, в якій  $x$ ,  $y$ ,  $z$  многочлени третього порядку (кубічні) відносно деякого параметра  $t$ . Оскільки ми розглядаємо кінцеві відрізки кривої, то без втрати загальності можемо обмежити діапазон зміни параметра і вважати  $0 \leq t \leq 1$ .

Отже

$$\begin{aligned} x(t) &= a_x \cdot t^3 + b_x \cdot t^2 + c_x \cdot t + d_x \\ y(t) &= a_y \cdot t^3 + b_y \cdot t^2 + c_y \cdot t + d_y \\ z(t) &= a_z \cdot t^3 + b_z \cdot t^2 + c_z \cdot t + d_z \end{aligned} \quad 0 \leq t \leq 1 \quad (2)$$

Похідні функцій  $x(t)$ ,  $y(t)$ ,  $z(t)$  по параметру  $t$  мають один і той же вигляд:

$$\frac{dx}{dt} = 3a_x t^2 + 2b_x t + c_x \quad (3)$$

Три похідні визначають дотичний вектор. Тангенси кутів нахилу кривої задаються як співвідношення компонент цього вектору:

$$\frac{dx}{dy} = \frac{dy/dt}{dx/dt}, \frac{dx}{dz} = \frac{dx/dt}{dz/dt} \text{ і т.д.} \quad (4)$$

Кути нахилу не залежать від довжини дотичного вектору. Якщо помножити похідні на  $k$ , отримаємо

$$k \frac{dx}{dt}, k \frac{dy}{dt}, k \frac{dz}{dt} \quad (5)$$

$$\text{Тоді } \frac{k \frac{dy}{dt}}{k \frac{dx}{dt}} = \frac{dy/dt}{dx/dt} = \frac{dy}{dx} \quad (6)$$

При пошуку способів визначення  $a_x, b_x, c_x, d_x$  нижче ми матимемо справу тільки з похідними  $x(t)$ . Для  $y(t)$  і  $z(t)$  похідні і остаточні формули аналогічні формулам для  $x(t)$  і ми їх не приводитимемо.

Чому ми розглядаємо саме кубічні криві? Тому, що для сегментів кривої не існує представлення нижчого порядку, яке б забезпечувало в точці з'єднання кривих один з одним безперервність положення і нахилу сегментів і в той же час гарантувало б, що кінцеві точки сегменту кривої проходять через задані точки.

Параметричний кубічний многочлен з чотирма коефіцієнтами є параметричною кривою найбільш низької міри, яка при відповідному виборі коефіцієнтів може задовольняти чотирьом умовам (положенню кожного з кінців сегменту і дотичним векторам в них).

Ми розглянемо тільки три способи опису параметричних бікубічних кривих: метод Ерміта, в якому задаються положення кінцевих точок кривої і дотичні вектори в них; метод Без'є, в якому задається положення кінцевих точок кривої, а для неявного завдання дотичних в цих точках використовуються дві інші точки, що зазвичай не лежать на цій кривій; метод В-сплайнів, при якому кінцеві точки не лежать на кривій, внаслідок чого як перша, так і друга похідна виявляються неперервними на кінцях сегменту. (Рис 4)

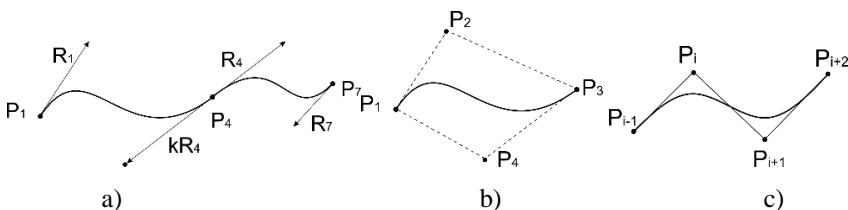


Рис. 2.2 – форми задання бікубічних кривих: а – Ерміта; б – Без'є, с – В-сплайн.

У формі Ерміта, задаються кінцеві точки і дотичні вектори до кривої в цих точках (рис. 2.2-а) Задамо точки  $P_1$  і  $P_4$  і дотичні вектори  $R_1$  і  $R_4$  (точкам надаються індекси 1 і 4, а не 1 і 2 для сумісності з виразами, які використовуються при побудові кривих методами Без'є і В-сплайнів). Вимагається знайти коефіцієнти  $a_x, b_x, c_x, d_x$  з виразу (2), що задовольняє умовам

$$x(0) = P_{1x}, x(1) = P_{4x}, x'(0) = R_{1x}, x'(1) = R_{4x} \quad (7)$$

Ми використовуємо індекс  $x$  для посилання на  $x$ -компоненти точок і дотичних векторів. Перепишучи вираження для  $x(t)$ , отримаємо.

$$x(t) = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, \text{ або } x(t) = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \cdot \bar{C}_x, \text{ або } x(t) = T \cdot C_x \quad (8)$$

де  $T$  є вектор-рядок степеней  $t$ , а  $C_x$  – вектор-стовпець коефіцієнтів  $x(t)$ .

Запишемо умови (7), використовуючи рівняння (8).

$$x(0) = P_{1x} = (0, 0, 0, 1) \cdot C_x, x(1) = P_{4x} = (1, 1, 1, 1) \cdot C_x \quad (9)$$

Щоб записати вираз для обмежень на дотичні вектори, продиференціюємо вираз (9) по  $t$  і отримаємо:

$$x'(t) = (3t^2, 2t, 1, 0) \cdot C_x, x'(0) = (0, 0, 1, 0) \cdot C_x, x'(1) = (3, 2, 1, 0) \cdot C_x \quad (10)$$

Вирази (9), (10), можна об'єднати в одне матричне рівняння

$$\begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \cdot \vec{C}_x \quad (11)$$

Знайшовши зворотню матрицю, ми отримаємо вираз для  $C_x$ , якщо задамо:

$$\vec{C}_x = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix}_x = \|M_h\| \cdot \|G_{hx}\| \quad (12)$$

Через  $\|M_h\|$  ( $M_h$ ) тут позначена ермітова матриця, а через  $\|G_h\|$  ( $G_h$ ) геометричний вектор Ерміта. Підставляючи цей результат у (8), отримаємо:

$$x(t) = T \cdot M_{hx} \cdot G_{hx}, y(t) = T \cdot M_{hy} \cdot G_{hy}, z(t) = T \cdot M_{hz} \cdot G_{hz} \text{ або} \\ P(t) = T \cdot M_h \cdot G_h \quad (13)$$

Якщо задані  $P_1, P_4, R_1, R_4$ , можна визначити  $x(t), y(t), z(t)$  для  $0 \leq t \leq 1$  і знайти усі точки на сегменті кубічної кривої від  $P_1$  до  $P_4$ , у якого дотичний вектор початкової точки дорівнює  $R_1$ , а в кінцевій –  $R_4$ .

$$T \cdot M_h = \|(2t^3 - 3t^2 + 1), (-2t^3 + 3t^2), (t^3 - 2t^2 + t), (t^3 - t^2)\| \quad (14)$$

Помноживши (14) на  $G_{hx}$ , отримаємо:

$$x(t) = T \cdot M_{hx} G_x = P_{1x}(2t^3 - 3t^2 + 1) + P_{4x}(-2t^3 + 3t^2) + R_{1x}(t^3 - 2t^2 + t) + R_{4x}(t^3 - t^2) \quad (15)$$

Чотири функції змінної  $t$  в вигляді (15) називають функціями сполучення, оскільки за допомогою двох перших сполучаються точки  $P_1$  і  $P_4$ , а за допомогою двох інших - вектори  $R_1$  і  $R_4$ , внаслідок чого досягається "згладжене" об'єднання  $x(t)$ .

Форма опису Без'є дуже близька до ермітової форми, але відрізняється заданням дотичних векторів в кінцевих точках.

Дотичні вектори в кінцевих точках задаються відрізками  $[P_1 P_2]$  і  $[P_3 P_4]$ .

Зокрема дотичні вектори ермітової форми  $R_1$  і  $R_4$  визначалися у такий спосіб, щоб відповідати чотирьом точкам Без'є  $P_1, P_2, P_3, P_4$ :

$$R_1 = 3(P_2 - P_1) = P'(0), R_4 = 3(P_4 - P_3) = P'(1)$$

Тому співвідношення між ермітовою матрицею і матрицею Без'є  $G_b$  в загальному випадку записується так:

$$G_h = \begin{vmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & -3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{vmatrix} \cdot \begin{vmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{vmatrix} = M_{hb} \cdot G_b \quad (16)$$

Підставляючи (16) в (13), знайдемо:  $x(t) = T \cdot M_h \cdot G_{hx} = T \cdot M_h \cdot M_{hb} \cdot G_{bx}$

Позначивши через  $M_b$  добуток  $M_h \cdot M_{hb}$ , отримаємо вираз

$x(t) = T \cdot M_b \cdot G_{bx}$ , який має тепер форму Без'є.  $x(t) = T \cdot M_b \cdot G_{bx}$

$$M_b = \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \quad (17)$$

Неперервність при сполученні двох кривих Без'є забезпечується дотриманням рівності векторів  $P_3P_4 = k P_4P_5$ .

Форма Без'є завдяки своїм двом властивостям використовується в машинній графіці частіше, ніж ермітова форма. По-перше, задавати (і змінювати) дотичні вектори у формі двох точок в інтерактивному режимі зручніше, ніж в явному виді. По-друге чотири управляючі точки, визначають опуклий багатокутник (*опуклу оболонку*), усередині якої знаходиться крива Без'є. Опукла оболонка виявляється корисною при відсіканні кривої по вікну або видимому об'єму.

Сплайн (spline) називали гнучку металеву лінійку — універсальне лекало, яке використовували для того, щоб гладко з'єднати окремі точки на кресленні, тобто для графічної інтерполяції. Більше того, крива, що описує деформацію гнучкої лінійки, зафіксованої в окремих точках, є сплайном. Таким чином гнучка лінійка є фізичною моделлю сплайн-функції (або, навпаки, сплайн-функція є математичною моделлю гнучкої лінійки).

В-сплайн (базисний сплайн) – сплайн-функція, що має мінімальний носій для заданого степеня, порядку гладкості та області визначення.

Фундаментальна теорема стверджує, що довільна сплайн-функція заданого степеня, гладкості і області визначення може бути представлена як лінійна комбінація В-сплайнів того ж степеня і гладкості на тій же області визначення.

Крива, представлена у вигляді кубічного В-сплайна, в загальному випадку може проходити через будь-які управляючі точки, проте вона неперервна і, крім того, неперервні її дотичний вектор і кривизна (тобто перша і друга похідні кривої неперервні в кінцевих точках) у відмінності від форм Ерміта і Без'є, у яких в кінцевих точках неперервні тільки перші похідні, які проходять через управляючі точки. Таким чином, можна стверджувати, що форма В-сплайнів більш "гладка", ніж інші форми. Термін сплайн походить від назви довгих гнучких металевих лінійок, за допомогою яких креслярі розмічали поверхні літаків і кораблів.



Металевий сплайн, якщо не сильно напружений, також має неперервність другого порядку. В-сплайн описується наступною формулою:

$$x(t) = T \cdot M_s \cdot G_{sx} \quad (18)$$

$$\text{де} \quad \|M_s\| = \frac{1}{6} \cdot \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{vmatrix} \quad (19)$$

При апроксимації управляючих точок  $P_1, P_2, \dots, P_n$  послідовністю В-сплайнів ми будемо застосовувати між кожною парою сусідніх точок геометричні матриці. Для апроксимації в інтервалі, близькому до точок  $P_i$  та  $P_{i+1}$ , (рис. 2.2-с) використовується

$$\bar{G}_s^i = \begin{pmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{pmatrix}, \quad 2 \leq i \leq n-2 \quad (20)$$

Властивість опуклої оболонки кривих Без'є також справедливою для кривих у формі В-сплайнів: опукла оболонка кривої в наближеному інтервалі  $P_i - P_{i+1}$  та ж сама, що і для чотирьох управляючих точок, використовуваних для генерації кривої:  $P_{i-1}, P_i, P_{i+1}, P_{i+2}$

Форма Ерміта придатна для апроксимації вже наявних поверхонь, коли необхідно добитися як відповідності точок, так і відповідності векторів, тоді як представлення у вигляді В-сплайнів зручно для апроксимації точок і досягнення  $C^{(2)}$ -неперервності. Форми Без'є і В-сплайнів придатні для роботи в інтерактивному режимі. Обидві ці форми мають властивість опуклої оболонки. Відмітимо, що криву, спочатку задану в одній формі, можна перетворити в іншу форму, якщо записати геометричний вектор першої форми в термінах другої.

### 2.3. Параметричні бікубічні поверхні

Перейдемо тепер від кубічних кривих до бікубічних поверхонь, що задаються кубічними рівняннями від двох змінних  $s$  і  $t$ . Змінюючи обидва параметри від 0 до 1, можна визначити усі точки на шматку поверхні. Якщо одному з параметрів присвоїти постійне значення, а іншій змінювати в діапазоні 0 – 1, то в результаті отримаємо кубічну криву. Так само як і у разі кривих, ми розглядатимемо рівняння тільки для  $x$ :

$$x(s, t) = a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t^4 + a_{14}s^3 + a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + \quad (1)$$

$$+ a_{24}s^2 + a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44}$$

$$\text{або в алгебраїчній формі:} \quad x(s, t) = S \cdot C_x \cdot T^T \quad (2)$$

де  $S = (s^3 \ s^2 \ s \ 1)$ ,  $T = (t^3 \ t^2 \ t \ 1)$ , а  $T^T$  означає транспоновану матрицю  $T$ .  $C_x$  задає коефіцієнти бікубічного многочлена. Існують також  $C_y$  і  $C_z$  які задають коефіцієнти  $y(s, t)$  і  $z(s, t)$ .

Щоб записати поверхню у формі Ерміта застосовується підхід, аналогічний викладеному для бікубічних кривих, що дозволяє використати управляючі точки, і дотичні вектори для визначення коефіцієнтів бікубічного многочлена.

Розглянемо рівняння бікубічної кривої п 2.1 (13), в якому  $t$  замінимо на  $s$ .

$$x(s) = S \cdot M_h \cdot G_{hx} \quad (3)$$

Перепишемо це рівняння так, щоб геометрична матриця Ерміта була не константою, а функцією  $t$ .

$$x(s, t) = S \cdot M_h \cdot G_{hx}(t) = S \cdot M_h \cdot \begin{pmatrix} P_1(t) \\ P_4(t) \\ R_1(t) \\ R_4(t) \end{pmatrix}_x \quad (4)$$

Функції  $P_{1x}(t)$  і  $P_{4x}(t)$  описують  $x$ -компоненти початковою і кінцевою точок кривої, що задається параметром  $s$ . Для кожного значення  $t$  визначаються деякі дві кінцеві точки. Аналогічно  $R_{1x}(t)$  і  $R_{4x}(t)$  описують дотичні вектори в кінцевих точках кубічної кривої, побудованої залежно від  $s$ .

## 2.4. Полігональні сітки

Полігональна сітка – це сукупність пов'язаних між собою плоских багатокутників. Недоліком цього методу є наближеність.

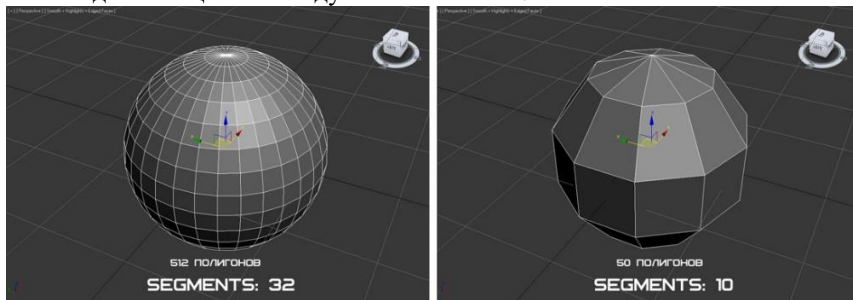


Рис. 2.3 – полігональна апроксимація сфери в 3dmax.

Можна збільшити міру наближення, збільшуючи число багатокутників. Але це веде до додаткових витрат пам'яті і часу роботи алгоритмів, що працюють з таким представленням Рис. 2.3

Полігональні сітки складаються з ребер, вершин і багатокутників. Вершини з'єднуються ребрами, а багатокутники розглядаються як послідовності ребер або вершин. Сітку можна представити декількома різними способами.

#### 2.3.1. Явне задання багатокутників

Кожен багатокутник представляється у вигляді списку координат його вершин :  $P = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots (x_n, y_n, z_n))$ .

Вершини задаються в тому порядку, в якому вони зустрічаються при обході. При цьому усі послідовні вершини багатокутника, а також перша і остання з'єднуються ребрами. Для кожного окремого багатокутника цей спосіб запису є ефективним, проте для полігональної сітки дає великі втрати пам'яті внаслідок дублювання інформації про координати спільних вершин. Більше того, немає явного опису спільних вершин. Наприклад, пошук усіх багатокутників, що мають загальну вершину, вимагає порівняння трійок координат вершин одного багатокутника з трійками координат вершин інших багатокутників.

#### 2.3.2. Задання багатокутників за допомогою покажчиків на список вершин

Кожен вузол полігональної сітки запам'ятовується лише один раз в списку вершин  $V = ((x_1, y_1, z_1), (x_2, y_2, z_2), \dots (x_n, y_n, z_n))$ .

У такому представленні вдається заощадити значну кількість пам'яті. Крім того координати вершин можуть легко змінюватися. Проте все ще непросто відшукувати багатокутники із загальними ребрами

#### 2.3.3. Явне задання ребер

У цьому представленні є список вершин  $V$ . Багатокутник розглядається як сукупність покажчиків на елементи списку ребер, в якому ребра зустрічаються лише один раз. Кожне ребро в списку ребер вказує на дві вершини в списку вершин, що визначають це ребро, а також на один або два багатокутники, якими це ребро належить. Таким чином, ми описуємо багатокутник як

$P = (E_1, \dots, E_n)$  а ребро, наприклад, як  $E = (V_1, V_2, P_1, P_2)$ .

Якщо ребро належить одному багатокутнику, то або  $P_1$  або  $P_2$  –пусте.

Полігональна сітка зображується шляхом викреслювання не усіх багатокутників, а усіх ребер. Окремі багатокутники при цьому також зображуються досить просто.

## **3. ЗОБРАЖЕННЯ ТРИВИМІРНИХ ОБ'ЄКТІВ**

### **3.1. Поняття про світові, віконні та екранні координати**

У двовірній графіці, якщо користувач збільшив зображення і на екрані залишилася видно тільки невелика частина, програма може заощадити процесорний час і пам'ять і не промальовувати ті частини зображення, які залишилися "за кадром".

Аналогічно, в тривимірній графіці, сцена може складатися з об'єктів, розташованих з усіх боків кадру віртуальної камери, але програмі достатньо прорахувати тільки ті об'єкти, які знаходяться в полі зору. У тривимірній графіці, це нетривіальна задача.

Координати реальної сцени називаються світовими. Об'єкт не завжди креслиться цілком. У комп'ютерній графіці визначається поняття вікна. Це область, що задана у світових координатах, у межах якої креслиться об'єкт. У найпростішому випадку – це прямокутник зі сторонами, що паралельні координатним осям. Важливо визначити, що вікно відноситься до об'єкта, а не до зображення, яке буде сформоване. Область на екрані, на яку відображується вікно називається областю виводу (полем виводу). Двовимірні координати поля виводу називаються екранними координатами. Оскільки найменший адресований елемент це піксель, екранні координати завжди приймають натуральні значення.

У двовимірному випадку задається вікно в двовимірному світовому координатному просторі та поле виведення на двовимірній видовій поверхні. У загальному випадку, об'єкти, описані у світових координатах, відсікаються по межі видимого об'єму, а після цього перетворюються в полі виведення для дисплея. Складність, характерна для тривимірного випадку, виникає тому, що видова поверхня не має графічного третього виміру.

Наприклад в графічній бібліотеці OpenGL відрізняють наступні координати: (рис. 3.1)

OCS - object coordinate system

WCS - world coordinate system

VCS - viewing coordinate system

CCS - clipping coordinate system

NDCS - normalized device coordinate system

DCS - device coordinate system

об'єктні координати;

світові координати;

видові координати;

відсічені координати;

нормалізовані екранні

екранні координати;

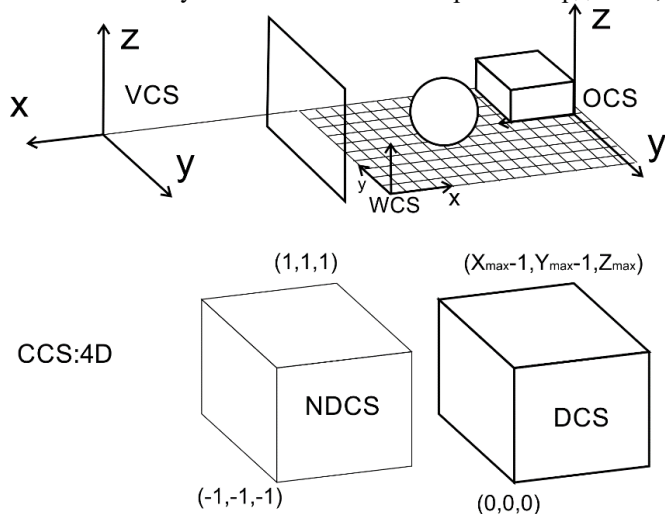


Рис.3.1 Системи координат у відповідності до графічного конвеєра OpenGL

### 3.2. Проекції

Невідповідність між просторовими об'єктами і плоскими зображеннями усуває проектування створення проекції тривимірних об'єктів на двовимірній проекційній площині. Проекція тривимірного об'єкту, представленого у вигляді сукупності точок, будується за допомогою прямих проєктуючих променів, які називаються проєкторами і виходять з центру проєкції, проходять через кожну точку об'єкта і, перетинаючи картинну площину, утворюють проєкцію. (Рис.3.2)

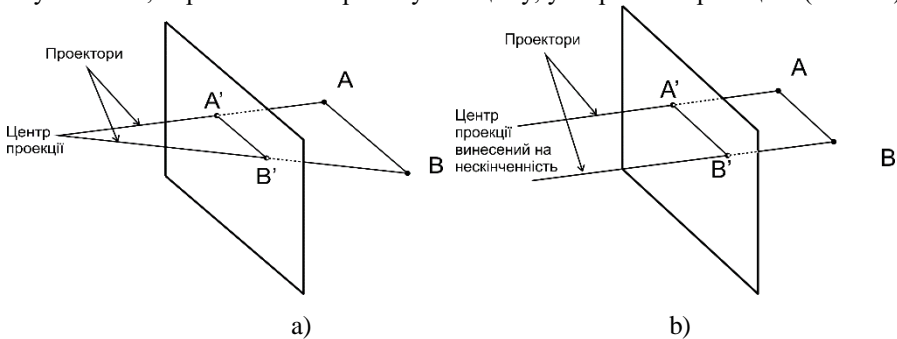


Рис. 3.2 – Відрізок АВ і його проєкція А'В': а – центральна; б – паралельна.

Оскільки проєкція відрізка сама є відрізком, то достатньо спроектувати лише кінцеві точки. Визначений таким чином клас проєкцій відомий під назвою плоских геометричних проєкцій, оскільки проектування в цьому випадку робиться на площину, а не на викривлену поверхню, і в якості проєкторів беруться прямі, а не криві лінії. У відмінності від них багато картографічних проєкцій є або неплоскими, або негеометричними.

Плоскі геометричні проєкції, які надалі називатимемо просто проєкціями, можна підрозділити на два основні класи: центральні і паралельні. Відмінність між ними визначається співвідношенням між центром проєкції і проєкційною площиною. Якщо відстань між ними кінцева, то проєкція буде центральною, якщо ж нескінченна, то проєкція буде паралельною. При описі центральної проєкції ми явно задаємо її центр проєкції, тоді як визначаючи паралельну проєкцію, ми вказуємо напрям проектування.

Центральна проєкція породжує візуальний ефект, аналогічний тому, до якого призводять фотографічні системи або зорова система людини, і тому використовується у разі, коли вимагається досягти певної міри реальності. Цей ефект відомий як перспективне укорочення: розмір центральної проєкції об'єкту зменшується з наближенням проєкційної площини до центру проєкції. Це означає, що, хоча центральна проєкція є реалістичною, вона виявляється непридатною для представлення точної форми і розмірів об'єкта: з проєкції не можна отримати

інформацію про відстані; кути зберігаються тільки на тих гранях об'єкту, які паралельні проекційній площині; проєкції паралельних ліній в загальному випадку не паралельні.

Паралельна проєкція породжує менш реалістичне зображення, оскільки відсутнє перспективне укорочення, хоча при цьому можуть мати місце різні постійні укорочення уздовж кожної з осей. Проєкція фіксує істинні розміри (з точністю до скалярного множника), і паралельні прямі залишаються паралельними. Як і у разі центральної проєкції, кути зберігаються тільки на тих гранях об'єкту, які паралельні проекційній площині.

Центральні проєкції будь-якої сукупності паралельних прямих, які не паралельні проекційній площині сходяться в точці сходження. Існує нескінченне число точок сходження. Якщо сукупність прямих паралельна одній з головних координатних осей, то їх точка сходження називається головною точкою сходження. Є тільки три такі точки, що відповідають трьом координатним осям. Якщо, наприклад, проєкційна площина перетинає одну тільки вісь  $z$  і, отже, перпендикулярна їй, то лише на цій осі лежатиме головна точка сходження, оскільки прямі, паралельні як  $y$ -, так і  $x$ -осям, паралельні також і проекційній площині і тому не мають точки сходження.

Центральні проєкції класифікуються залежно від числа головних точок сходження, які вони мають, а отже, і від числа координатних осей, які перетинає проєкційна площина.

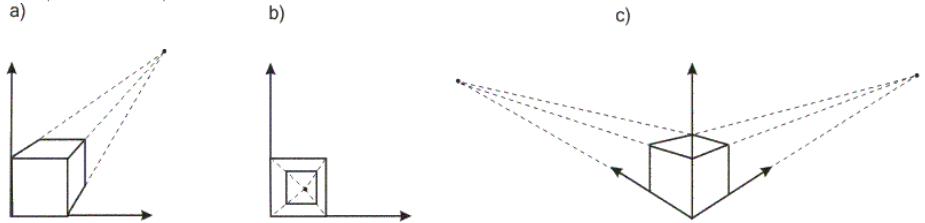


Рис. 3.3 – Центральні проєкції куба на площину: а), б) – одноточкові; в) двоточкова.

Двоточкова центральна проєкція широко застосовується в проектуванні і в рекламних зображеннях, в яких вертикальні прямі проєктуються як паралельні і, отже, не сходяться. Триточкові центральні проєкції майже зовсім не використовуються, по-перше, тому, що їх важко будувати, а по-друге, через те, що вони додають мало нового з точки зору реалістичності в порівнянні з двоточною проєкцією.

Паралельні проєкції розділяються на два типи залежно від співвідношення між напрямом проектування і нормаллю до проекційною площиною. у ортографічних паралельних проєкціях ці напрями співпадають, а в косокутних паралельних проєкціях вони не співпадають. Тобто в ортографічних проєкціях напрям проектування є нормаллю до проекційної площини.

Найбільш поширеними видами ортографічних проєкцій є вигляд спереду, вигляд зверху(план) і вигляд збоку, (рис. 3.4), в яких картинна площина перпендикулярна головним координатним осям, співпадаючим внаслідок цього з напрямом проєктування. По цих проєкціях можна вимірювати відстані і кути.

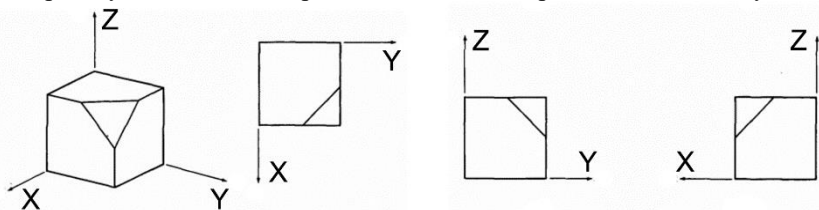


Рис. 3.4 – Ортографічні проєкції усіченого куба

У разі аксонометричних ортографічних проєкцій використовуються проєкційні площини, не перпендикулярні головним координатним осям, тому на них зображуються відразу декілька сторін об'єкта, так само як і при центральному проєктуванні. Проте в аксонометрії укорочення постійне, тоді як для центральної проєкції воно пов'язане з відстанню від центру проєкції. При аксонометричному проєктуванні зберігається паралельність прямих, а кути змінюються; відстані можна вимірювати уздовж головних координатних осей(у загальному випадку, з різними масштабними коефіцієнтами).

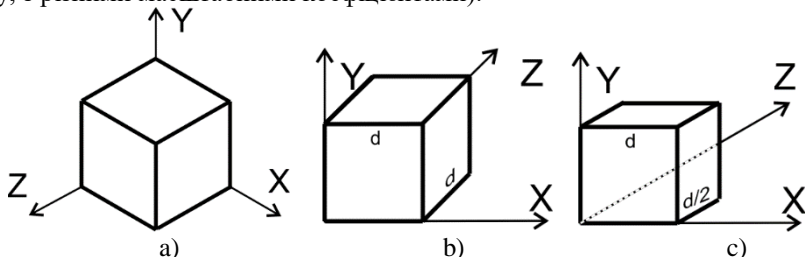


Рис. 3.5 – Проекції одиничного куба: а) – ізометрична; б) – кавале; с) – кабінє.

Часто використовувана аксонометрична проєкція є ізометрична проєкція (рис.3.5-а). В цьому випадку нормаль до проєкційної площини(а отже і напрям проєктування) складає рівні кути з кожною з головних координатних осей. Ізометрична проєкція має наступну властивість: усі три головні координатні осі однаково коротшають. Тому можна проводити виміри уздовж напрямку осей з одним тим же масштабом (звідси назва ІЗО-дорівнює).

Косокутні проєкції. Проекційна площина перпендикулярна головній координатній осі. Сторона об'єкта, паралельна цій площині, проєктується так, що можна вимірювати кути і відстані. Проектування інших сторін також допускає проведення лінійних(але не кутових) вимірів уздовж головних осей.

Двома важливими видами косокутних проєкцій є проєкції *кавале* (рис. 3.5-b) – cavalier (у вітчизняній літературі горизонтальна косокутна ізометрія – військова перспектива) і *кабіне* (рис. 3.5-с) – cabinet (фронтальна косокутна диметрія – кабінетна проєкція).

У проєкції *кавале* напрям проєктування, як правило, складає з проєкційною площиною кут  $45^\circ$ . Відстані по кожній осі відкладаються однаково, що дозволяє виконувати виміри, але спотворює уявлення про реальну форму об'єкта.

Проєкція *кабіне* має напрям проєктування, який складає з проєкційною площиною кут  $\arccos(\frac{1}{2})$ . При цьому відрізки, перпендикулярні проєкційній площині, після проєктування складають половину їх дійсної довжини. Проєкції кабіні є реалістичнішими, ніж проєкції кавале, оскільки укорочення з коефіцієнтом  $\frac{1}{2}$  більше узгоджується з нашим візуальним досвідом.

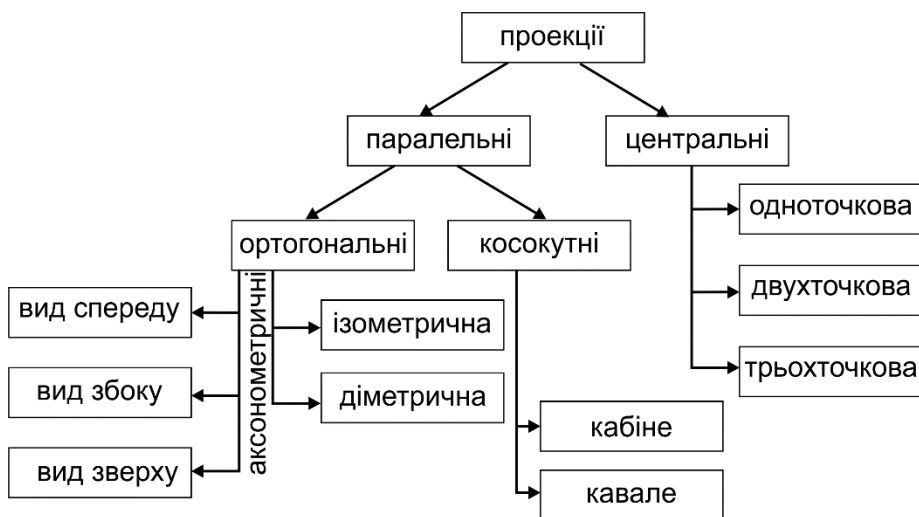


Рис. 3.6 – Види проєкцій

### 3.3. Математичний опис плоских геометричних проєкцій

Для простоти вважатимемо, що при центральному проєктуванні центр проєкції знаходиться на початку координат, картинна площина перпендикулярна осі  $Z$  і співпадає з площиною  $z = d$ , а при паралельному проєктуванні співпадає з площиною. Проєкції розглядаються в системі координат спостерігача, яка є лівобічною. Система координат, в якій вісь  $X$  спрямована вправо, вісь  $Y$  - вгору, а вісь  $Z$  - углиб екрану, природно узгоджується з екраном дисплея. Для переходу



від правосторонньої системи координат, в якій описувалися перетворення перенесення, повороту і масштабування до лівобічної, в якій описані проекції, можна скористатися матрицею

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ де координата } z \text{ множиться на } -1$$

Кожну з проекцій можна описати матрицею  $4 \times 4$ . Це виявляється зручним, оскільки з'являється можливість об'єднати матрицю проектування з матрицею афінних перетворень, представивши в результаті дві операції перетворення (геометричний-перенос, масштабування, поворот і проектування) у вигляді однієї матриці.

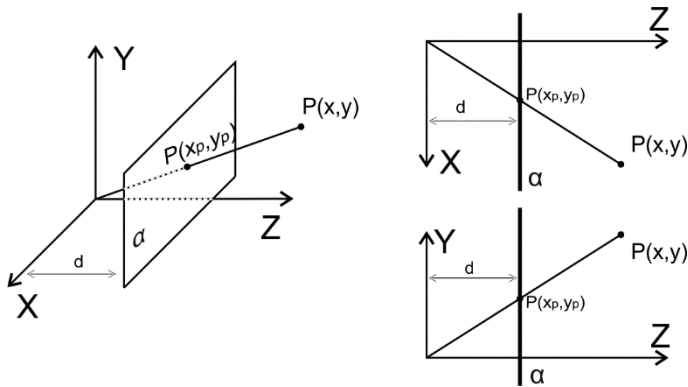


Рис. 3.7 – Проектування точки  $P$  на площину  $\alpha$ .

На рис. 3.7 приведено три зображення в лівобічній системі координат, в яких точка  $P$  проектується на проекційну площину, розташовану на відстані  $d$  від початку координат.

Для обчислення координат проекції точки  $(x, y, z)$  напишемо співвідношення, отримані з подібних трикутників :

$$\frac{x_p}{d} = \frac{x}{z}, \frac{y_p}{d} = \frac{y}{z} \quad (1)$$

$$x_p = \frac{d \cdot x}{z} = \frac{x}{\frac{z}{d}}, y_p = \frac{d \cdot y}{z} = \frac{y}{\frac{z}{d}} \quad (2)$$

Відстань  $d$  є в даному випадку масштабним множником, застосованим до координат  $x_p$  і  $y_p$ . Чинником, що призводить до того, що на центральній проекції більше видалені об'єкти виглядають дрібніше, ніж ближні, є ділення на  $z$ . Допустимі усі значення  $z$ , окрім  $z = 0$ . Точки можуть розташовуватися як позаду центру проекції на негативній частині осі  $Z$ , так і між центром проекції і проекційною площиною.

Це перетворення можна представити у вигляді матриці  $4 \times 4$  :

$$M_{\text{центр}} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{d} \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad (3)$$

Множивши точку  $P = |x \ y \ z \ 1|$  на матрицю  $M_{\text{центр}}$ , отримаємо загальний вираз для точки в однорідних координатах  $|X \ Y \ Z \ W|$ :

$$|X \ Y \ Z \ W| = P \cdot M_{\text{центр}} = |x \ y \ z \ 1| \cdot \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{d} \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad (4)$$

або

$$|X \ Y \ Z \ W| = |x \ y \ z \frac{z}{d}| \quad (5)$$

Тепер, поділивши на  $W$  (яке рівне  $\frac{z}{d}$ ) для зворотного переходу до трьох вимірів, маємо

$$\left| \frac{x}{W} \ \frac{y}{W} \ \frac{z}{W} \ 1 \right| = |x_p \ y_p \ z_p \ 1| = \left| \frac{x}{\frac{z}{d}} \ \frac{y}{\frac{z}{d}} \ d \ 1 \right| \quad (6)$$

$$|X / W \ Y / W \ Z / W \ 1| = |x_p \ y_p \ z_p \ 1| = |x / (z / d) \ y / (z / d) \ d \ 1|$$

Цей результат є коректним, оскільки містить перетворену  $z$ - координату, що відповідає положенню проєкційної площини уздовж осі  $Z$ .

Ортографічне проєктування на площину  $z = 0$  очевидне. Напрямок проєктування співпадає з нормаллю до площини проєкції, тобто в нашому випадку з віссю  $Z$ . Таким чином, точка  $P$  має координати:

$$x_p = x, y_p = y, z_p = 0 \quad (7)$$

Ця проєкція описується матрицею

$$M_{\text{орт}} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (8)$$

При косокутній проєкції, на площину  $XOY$  орт  $\bar{e}_3 = (0,0,1)$  переходить в  $\bar{r}_0 = (a, b, 0)$  тобто напрямок проєкції задається вектором  $\bar{p} = \bar{r}_0 - \bar{p}_3 = (a, b, -1)$  Таке перетворення в однорідних координатах можна задати з допомогою матриці

$$P = \begin{vmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (9)$$

В проєкції кавале вектор  $\bar{e}_3$  переходить у  $(\cos(\pi/4), \cos(\pi/4), 0)$ , а в проєкції кабіне  $(\frac{1}{2} \cos(\frac{\pi}{4}), \frac{1}{2} \cos(\frac{\pi}{4}), 0)$ .

## 4. РЕНДЕРИНГ

### 4.1. Математичні засади рендерингу

Рендеринг, комп'ютерна візуалізація (від англійського слова rendering — візуалізація, вимальовування, подання) — це процес отримання зображення за двовимірною або тривимірною моделлю (файлом сцени). Слово рендеринг вживають для позначення процесу візуалізації, що виконується за допомогою програмного забезпечення, а рендер — для позначення готового зображення, тобто як синоніми до словосполучень: комп'ютерна візуалізація — комп'ютерний рендеринг, візуалізований об'єкт — рендер.

На даний момент розроблено багато алгоритмів візуалізації. Існуюче програмне забезпечення може використовувати одночасно декілька алгоритмів для отримання кінцевого зображення. Спостерігається чіткий тренд на реалізацію алгоритмів рендерингу на апаратному рівні.

Ключем до теоретичного обґрунтування моделей рендерингу служить рівняння рендерингу. Воно є найповнішим формальним описом частини рендерингу, що не відноситься до сприйняття кінцевого зображення. Усі моделі є деяким наближеним рішенням цього рівняння.

$$L_o(x, \vec{w}) = L_e(x, \vec{w}) + \int_{\Omega} f_r((x, \vec{w}', \vec{w})) L_i(x, \vec{w}') (\vec{w}' \cdot \vec{n}) d\vec{w}'$$

Фізичною основою рівняння є закон збереження енергії. Неформальне тлумачення таке: кількість світлового випромінювання ( $L_o$ ), що виходить з певної точки в певному напрямку є власне випромінювання і відбите випромінювання. Відбите випромінювання є сума по всіх напрямках випромінювання, що надходить ( $L_i$ ), помноженого на коефіцієнт віддзеркалення з даним кутом. Об'єднуючи в одному рівнянні яке надходить, з тим, що випромінюється, в одній точці, це рівняння формує опис усього світлового потоку в заданій системі. Різноманітні алгоритми комп'ютерної графіки розв'язують це основне рівняння.

### 4.2. Основні прийоми рендерингу

Трасування кожного променя світла в сцені непрактично і займає неприйнятно довгий час. Навіть трасування малої кількості променів, достатньої, щоб отримати зображення, займає надмірно багато часу, якщо не застосовується апроксимація або семплування.

Семплування прийом частіше застосований в електронній музиці, полягає в поділі треку на повторювані частини семпли із яких можна створювати нові твори шляхом модуляцій і повторів.

Сьогодні в основному використовується, чотири групи методів, ефективніших, ніж моделювання усіх променів світла, що освітлюють сцену:

**1. Растеризування** — це переклад зображення, описаного векторним форматом в пікселі або точки, для виведення на дисплей або принтер. Геометрично проектує об'єкти на площину зображення, без врахування оптичних ефектів.

Можна виділити алгоритми растеризування двовимірних сцен, та алгоритми скануючої лінії (Scanline) який функціонує в режимі «рядок за рядком». Припустимо, що нам необхідно отримати зображення роздільної здатності  $1024 \times 1080$ . Програма проводить уявну лінію через кожен піксель, обчислюючи, які полігони лежать на шляху цієї лінії і обчислює необхідний колір пікселя, залежно від того, які текстури і кольори були призначені полігонам, що зустрілися на шляху цієї уявної лінії. Потім береться наступний піксель, потім наступний, – і так до кінця. Перевага методу в тому, що немає необхідності транслювати координати всіх вершин з основної до робочої пам'яті — лише вершини, що потрапляють в зону видимості. Кожна з вершин зчитується до робочої пам'яті лише раз, що значно підвищує швидкість виконання рендерингу.

Основною проблемою цього способу є відділення видимих полігонів від невидимих. У перших поколіннях такого роду рендерерів процес обчислення полігонів розпочинався з найдальшої точки від глядача і кожен новий полігон зафарбовував попередній. Такий підхід далекий від ідеалу із-за великої кількості непотрібних операцій. Щоб розв'язати цю проблему використовується  $Z$ —буфер. Програма обчислює усі полігони, що лежать на шляху уявної лінії і призначає кожному полігону  $Z$ —значення залежно від його віддаленості від екрану. Коли настає час рендерингу, обчислюються тільки полігони з найменшим  $Z$ —значенням — інші просто відкидаються.

**2. Кидання променів** (рейкастинг від англійського ray casting). Сцена розглядається, як така, що спостерігається з певної точки. З точки спостереження на об'єкти сцени направляються промені, за допомогою яких визначається колір пікселя на двовимірному екрані. При цьому промені припиняють своє поширення (на відміну від методу зворотного трасування), коли досягають будь-якого об'єкта сцени або її фону. Можливе використання будь-яких дуже простих способів додавання оптичних ефектів. Ефект перспективи отримується природним чином, якщо промені надходять під кутом, що залежить від положення пікселя на екрані і максимального кута об'єктиву камери.

**3. Трасування променів** (рейтрейсинг від англійського ray tracing) схоже на метод кидання променів. З точки спостереження на об'єкти сцени спрямовуються промені, за допомогою яких визначається колір пікселя на двовимірному екрані. Але при цьому промінь не припиняє своє поширення, а розділяється на три компоненти, кожен з яких вносить свій вклад в колір пікселя на двовимірному екрані: відбитий, тінювий і заломлений. Кількість таких розділень на компоненти визначає глибину трасування і впливає на якість і реалістичність зображення.

**4. Трасування шляху** (пачтрейсинг від англійської path tracing) містить схожий принцип трасування поширення променів, проте цей метод є найнаближеним до фізичних законів поширення світла. Також є самим ресурсоемним.

Сучасне програмне забезпечення зазвичай поєднує в собі декілька технік, щоб отримати досить якісне і фотореалістичне зображення за прийнятні витрати обчислювальних ресурсів.

## 5. ЗГЛАДЖУВАННЯ ЗОБРАЖЕНЬ

### 5.1. Причини і види спотворень зображення

Згладжування (англійською *anti-aliasing* або *smoothing*) — технологія, що використовується в обробці зображень з метою зробити межі кривих ліній більше гладкими, прибираючи «зубці» (рис. 5.1-а), що виникають на краях об'єктів або «розмиття» зображення при збільшенні його роздільної здатності (рис. 5.1-б). Вперше згладжування було застосовано в 1972 в Массачусетському технологічному інституті в групі *Architecture Machine*, яка пізніше стала основною частиною *MIT Media Lab*.

Оскільки растрове зображення дискретне по своїй природі існують різні види спотворень, наприклад:

- драбинний ефект (рис 5.1-а);
- некоректна візуалізація тонких і малих деталей; (рис 5.2);
- мерехтіння при русі дрібних об'єктів (рис 5.3).

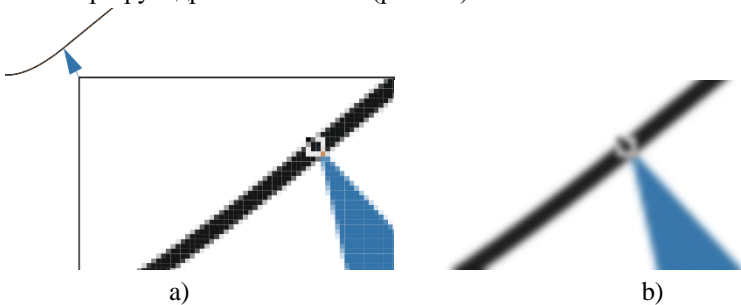


Рис. 5.1 – Приклади застосування згладжування: а) – усунення драбинного ефекту; б) – розмиття того ж зображення після збільшення роздільної здатності

Причина виникнення драбинного ефекту полягає у тому, що відрізки, ребра многокутників мають один атрибут кольору, кольорові межі та інше мають безперервну природу, тоді як растровий пристрій дискретний.

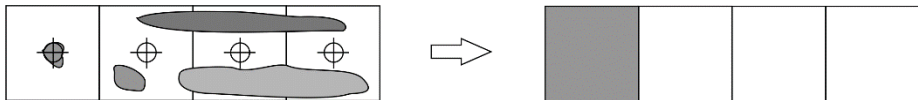


Рис. 5.2 – Некоректна візуалізація дрібних об'єктів

Якщо для визначення атрибутів використовується середина пікселя, на лівому пікселі (рис 5.2) весь піксель матиме атрибути маленького об'єкта. На правих пікселях – ілюструються об'єкти, які будуть проігноровані або втрачені. При цьому можуть бути проігноровані і довгі тонкі об'єкти.

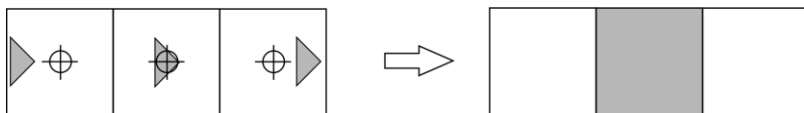


Рис. 5.3 – Мерехтіння дрібного об'єкта, що рухається

Особливо помітні такі ефекти в анімаційній послідовності кадрів. На рис 5.3 представлений маленький трикутник на трьох кадрах анімаційної послідовності. Якщо атрибути пікселя визначаються в його центрі, то на першому кадрі об'єкт невидимий, на другому – видимий, а на третьому – знову невидимий. В сукупності це виглядатиме як мерехтіння об'єкта.

## 5.2 Основні методи усунення спотворень

Перший метод пов'язаний зі збільшенням роздільної здатності растру, що обмежується можливостями конкретного дисплея. Таке обмеження передбачає, що растр необхідно обчислювати з більш високою роздільністю, а зображувати з меншою, використовуючи певне усереднення для отримання атрибутів пікселя з нижчою роздільністю (усереднення може бути рівномірне або зважене).

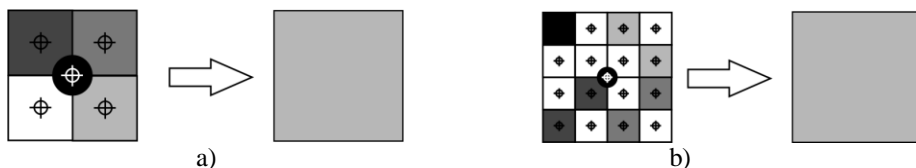


Рис. 5.4 – Рівномірне усереднювання кольору навколишніх пікселів для зменшення роздільної здатності по кожній координаті а) – в 2 рази ; б) – в 4 рази

На рис. 5.4 показано усереднювання двох типів. Кожен дисплейний піксель ділиться на підпікселі і формується растр більш високої роздільної здатності. Для отримання атрибутів дисплейного пікселя визначаються атрибути в центрі кожного підпікселя, які потім усереднюються. В деякій мірі можна отримати кращі результати, якщо розглядати більше підпікселів і враховувати їх вплив за допомогою вагів при визначенні атрибутів. На рис. 5.5 представлені зважені середні, запропоновані Франкліном Кроу, для зменшення роздільної здатності в 2 і 4 рази. Для цих зважених середніх при збільшенні роздільності в 2 рази розглядаються 9 підпікселів, а при зменшенні в 4 рази - 49.

1	2	1
2	<b>4</b>	2
1	2	1

1	2	3	4	3	2	1
2	4	6	8	6	4	2
3	6	9	12	9	6	3
4	8	12	<b>16</b>	12	8	4
3	6	9	12	9	6	3
2	4	6	8	6	4	2
1	2	3	4	3	2	1

Рис. 5.5 – Зважене усереднення характеристик пікселя за Ф. Кроу для зменшення роздільної здатності в 2 і 4 рази по кожній координаті

При цьому з найбільшою вагою враховується підпіксель, що відповідає центру дисплейного пікселя.

Наступний метод усунення спотворень полягає у тому, щоб трактувати піксель не як точку, а як скінченну область. Що еквівалентно постфільтрації зображення.

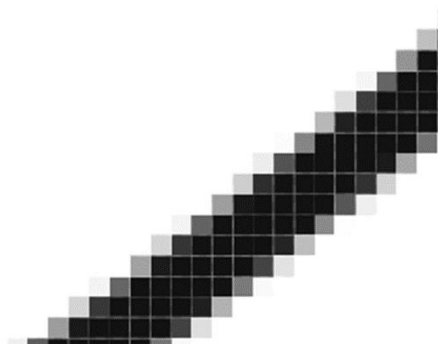


Рис. 5.6 – Згладжування драбинного ефекту шляхом використання відтінків сірого.

За наявності декількох інтенсивностей, або відтінків кольору, зовнішній вигляд ребра або відрізка може бути полішшений розмиванням країв. Простий евристичний метод полягає в тому, щоб встановлювати інтенсивність пікселя на ребрі пропорційно площі частини пікселя, що знаходиться у середині багатокутника.

### 5.3 Математичні основи роботи фільтрів

Узагальнення розглянутих простих методів згладжування вимагає використання математичного методу, званого згорткою. Для згладжування береться згортка сигналу, тобто зображення, з ядром згортки, а результат використовується для визначення атрибутів пікселя. Згортка задається інтегралом

$$c(\xi) = \int_{-\infty}^{+\infty} h(\xi - x) \cdot y(x) dx, \text{ де}$$

$h(\xi - x)$  – ядро або функція згортки,

$y(x)$  – згортання функції,

$c(\xi)$  – згортка  $h(\xi - x)$  і  $y(x)$ .

Уявити собі фізичний зміст згортки з її математичного визначення велими важко. Однак нам в цьому допоможе простий графічний аналіз.

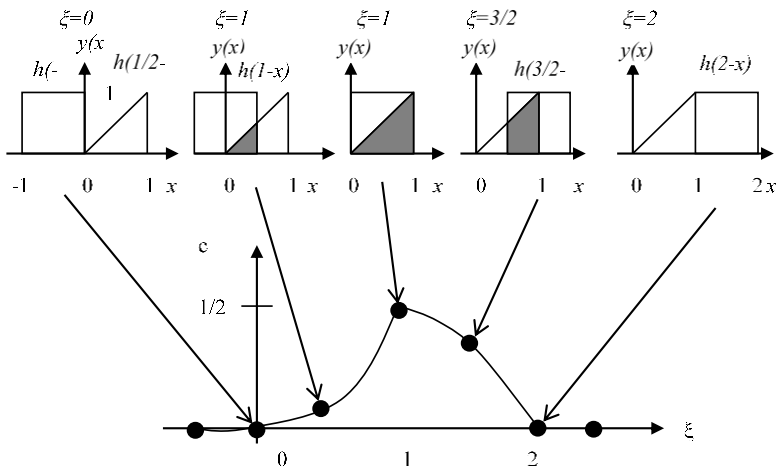


Рис. 5.7 – Згортка функції  $y(x)=x, 0 \leq x \leq 1$  з прямокутною функцією  $h(x)=1, 0 \leq x \leq 1$ .

Розглянемо згортку функції  $y(x) = x, 0 \leq x \leq 1$  простою прямокутною функцією згортки  $h(x) = 1, 0 \leq x \leq 1$ . У графічному вигляді функція згортки показана на (рис. 5.7). Для того, щоб отримати  $h(-x)$ , функцію треба відобразити відносно ординати (рис. б). Відображене ядро для отримання  $h(\xi - x)$  переноситься вправо на величину  $\xi$  (рис. с). Потім, як показано на нижньому малюнку, ядро згортки і згортання функції  $y(x)$  (рис. d) перемножують для різних значень величини  $\xi$ . Площа під об'єднаними кривими (функціями) дорівнює значенню згортки  $c(\xi)$ , яка теж зображена на нижньому малюнку. Зауважимо, що згортка не дорівнює нулю тільки в діапазоні  $0 \leq x \leq 2$  (або  $\xi$  в цьому діапазоні). Таким чином визначення згортки еквівалентне відображенню ядра згортки, його зрушенню, перемноженню двох функцій і визначенню площі, укладеної під об'єднаними кривими.



Щоб зрозуміти, який зв'язок існує між згорткою і усуненням ступінчастості (згладжуванням), згадаємо евристичний метод модуляції інтенсивності, в якому для визначення інтенсивності використовується площа пікселя. Перевірка функції згортки  $c(\xi)$  показує, що для  $m \leq 1$  значення згортки на правій стороні пікселя, тобто при  $x = \xi = 1$ , дорівнює тій частині пікселя, що знаходиться всередині багатокутника, тобто  $\frac{m}{2}$  (рис. б з  $y_i = 0$  з простого алгоритму усунення ступінчастості). Для  $m > 1$  значення згортки дає суму площ частин двох пікселів, що пересікаються, розташованих усередині багатокутника (рис. б с  $y_i = 0$  там же). Цей результат легко узагальнити на випадок  $y_i \neq 0$ . Таким чином, два попередні алгоритми (евристичне модулювання за площею і модифікований Брезенхема) еквівалентні згортці функцій ребер, тобто прямій  $y = m \cdot x + b$  і ядра згортки, обчислених на правій стороні пікселя.

Операція згортки часто називається фільтрацією, а ядро згортки функцією фільтра. В обговорюваному перш простому методі з площами зображення попередньо фільтрується. Префільтрація вирівнює атрибути пікселів обчисленого дозволу до виведення зображення на екран. Метод обчислення зображення з роздільною здатністю, більшим, ніж здатність дисплея, і подальше усереднення атрибутів декількох пікселів для отримання пікселів з меншим дозволом можна розглядати як постфільтруючу операцію.

Хоча за допомогою простого прямокутного фільтра або ядра згортки отримують цілком прийнятні результати, трикутний і гаусовський фільтри призводять до ще більш якісних результатів. Використовуються також двовимірні фільтри. В даний час досліджено просте прямокутне, пірамідалне кінцеве і двовимірне гаусовське ядра згортки або функції фільтра. Прості фільтри у вигляді згортки не завжди ефективні для маленьких багатокутників з площею, меншої площі пікселя або ж довгих тонких багатокутників. Проте згладжування можна реалізувати за допомогою відтинання. Сторони області пікселя утворюють відтинаюче вікно. Кожен індивідуальний багатокутник відсікається по сторонах цього вікна. Для модулювання інтенсивності пікселя використовується відношення площі отриманого в результаті відсікання багатокутника до площі пікселя. Якщо всередині пікселя знаходиться кілька багатокутників, то використовується середнє (або рівномірне, або зважене) їх атрибутів для модулювання атрибутів пікселя.

## СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Глава 33. Вычислительная геометрия // Алгоритмы: построение и анализ = Introduction to Algorithms. — 2-е издание. — М.: «Вильямс», 2005.
2. Ласло М. Вычислительная геометрия и компьютерная графика на C++. — М.: БИНОМ, 1997. — 304 с.
3. Петров, М. Н. Компьютерная графика : [учебное пособие для студ. вузов, обуч. по напр. подготовки дипломированных специалистов "Информатика и вычислительная техника"] / М. Н. Петров .— 3-е изд. — Санкт-Петербург : Питер, 2011 .— 544 с. : ил.
4. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение = Computational Geometry An introduction. — М.: Мир, 1989. — 478 с.
5. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. - М.:Мир, 1990. - 480 с.илл.
6. Фоли Дж., вэн Дэм А. Основы интерактивной машинной графики: В 2-х кн. Пер. с англ. - М.: Мир, 1985.

## ЗМІСТ

ВСТУП.....	3
1. АФІННІ ГЕОМЕТРИЧНІ ПЕРЕТВОРЕННЯ .....	4
1.2 Однорідні координати та.....	5
матричне представлення двовимірних перетворень .....	5
2. ПРЕДСТАВЛЕННЯ ПРОСТОРОВИХ ФОРМ.....	11
2.1. Способи представлення.....	11
2.2.Параметричні бікубічні криві .....	12
2.3. Параметричні бікубічні поверхні.....	17
3. ЗОБРАЖЕННЯ ТРИВИМІРНИХ ОБ'ЄКТІВ .....	19
3.1. Поняття про світові, віконні та екранні координати.....	19
3.2. Проекції .....	21
3.3. Математичний опис плоских геометричних проєкцій .....	24
4. РЕНДЕРИНГ.....	27
4.1. Математичні засади рендерингу .....	27
4.2. Основні прийоми рендерингу .....	27
5. ЗГЛАДЖУВАННЯ ЗОБРАЖЕНЬ .....	29
5. 1. Причини і види спотворень зображення .....	29
5.2 Основні методи усунення спотворень.....	30
5.3 Математичні основи роботи фільтрів .....	31

Навчальне видання

**Сирота Сергій Вікторович**

**Ліскін Вячеслав Олегович**

## **ОСНОВИ ОБЧИСЛЮВАЛЬНОЇ ГЕОМЕТРІЇ**

**Конспект лекцій**

**Дисципліни 03.14 ОБЧИСЛЮВАЛЬНА ГЕОМЕТРІЯ ТА  
КОМП'ЮТЕРНА ГРАФІКА  
для напряму підготовки  
6.040301 – «Прикладна математика»**

Відповідальний за випуск *Сирота С.В.*

Відповідальний редактор Копичко С.М.

Промислово-торговельна фірма «Просвіта»  
у формі товариства з обмеженою відповідальністю.

01032, Київ, бульвар Т. Шевченка, 48,

тел. (044) 234-15-86, 234-95-23 (факс).

Свідоцтво суб'єкта видавничої справи ДК № 221 від 16.10.2000 р.

Підп. до друку **03.11.15**. Формат 60x84<sup>1</sup>/<sub>16</sub>. Папір офс. Гарнітура Times.

Спосіб друку – різнограф. Ум. друк. арк. 1,85. Обл.-вид. арк. 1,0 Зам. №

Наклад 130 пр.

---

НТУУ «КПІ» ВПІ ВПК «Політехніка»

03056, Київ, вул. Політехнічна, 14, корпус 15

тел./факс (044) 406-81-78